CLAIMS

What is claimed is:

1. A method for revising code, comprising:

providing a first code module containing a resource;

transferring the resource from the first code module to a second code module;

providing forwarding information in a new version of the first code module that points to the second code module; and

accessing the resource in the second code module via the forwarding information in the first code module.

2. The method according to claim 1, wherein the first code module and second code module include resources in the form of type definitions that are accessible to an application program.

3. The method according to claim 1, wherein, subsequent to the transferring, the first code module includes no resources.

4. The method according to claim 1, wherein, subsequent to the transferring, the first code module includes remaining resources.

5. The method according to claim 1, wherein the forwarding information contains parameters for specifying a name of the resource and a location of the resource.

6. The method according to claim 1, wherein the forwarding information contains parameters for specifying a previous name of the resource, and a new name of the resource.

7. The method according to claim 1, further comprising repeating the above-referenced resource transferring and providing of the forwarding information one or more times to establish a chain of forwarding information, wherein the resource is accessed through the chain of forwarding information.

8. The method according to claim 1, wherein the accessing comprises compiling the first code module and the second code module to produce metadata that describes how the first code module is coupled to the second code module via the forwarding information, wherein the metadata is used to retrieve the resource from the second code module via the forwarding information.

9. The method according to claim 8, wherein the accessing further comprises generating machine-executable code based on the metadata.

10. A computer readable medium including machine readable instructions for implementing each of the providing a first code module, transferring, providing forwarding information, and accessing recited in claim 1.

11. An apparatus for compiling code, comprising:

a language compiler configured to accept at least a first code module and a second code module, wherein the first code module includes forwarding information that points to a resource in the second code module; and

wherein the language compiler includes logic configured to generate metadata that expresses a link between the first code module and the second code module based on the forwarding information in the first code module.

12. The apparatus according to claim 11, wherein the first code module and the second code module include resources in the form of type definitions that are accessible to an application program.

13. The apparatus according to claim 11, wherein the first code module includes no resources.

14. The apparatus according to claim 11, wherein the first code module includes additional resources other than the resource provided in the second code module.

15. The apparatus according to claim 11, wherein the forwarding information contains parameters for specifying a name of the resource and a location of the resource.

16. The apparatus according to claim 11, wherein the forwarding information contains parameters for specifying a previous name of the resource, and a new name of the resource.

17. The apparatus according to claim 11, wherein the compiler produces intermediate code that includes the metadata, wherein the apparatus includes another compiler configured to convert the intermediate code and the metadata into native code for execution on a machine.

18. A computer readable medium including machine readable instructions for implementing the compiler recited in claim 11.

19. A method for revising code, comprising:

providing a code module containing a resource;

providing renaming information in the code module that specifies a previous name of the resource, and a new name of the resource;

accessing the resource in the code module via the renaming information in the code module.

20. An apparatus for compiling code, comprising:

a language compiler configured to accept a code module containing a resource, wherein the code module includes renaming information that specifies a previous name of the resource, and a new name of the resource;

wherein the language compiler includes logic configured to generate metadata that expresses a change of name from the previous name to the new name based on the renaming information.

21. An apparatus for generating code, comprising:

logic configured to:

receive intermediate language code and metadata generated by a language compiler, wherein the metadata expresses a link between a first code module and a second code module, wherein the link identifies a location of a resource that was moved from the first code module to the second code module; and

process the intermediate language code and the metadata to generate processed code by automatically tracing the link.

22. An apparatus for generating code, comprising:

logic configured to:

receive intermediate language code and metadata generated by a language compiler, wherein the metadata expresses a link between an original name of a resource and a new name of the resource; and

process the intermediate language code and the metadata to generate processed code by automatically tracing the link.